BAB II
TINJAUAN PUSTAKA

2.1. Teori Umum

2.1.1. Pengertian Komputer

Asropudin (2013:19), "komputer adalah alat bantu pemrosesan data secara elektronik dan cara pemrosesan datanya berdasarkan urutan instruksi atau program yang tersimpan dalam memori masing-masing komputer".

Hamacher et.al (2010:1), "komputer adalah mesin penghitung elektronik yang cepat dan dapat menerima informasi input digital, kemudian memprosesnya sesuai dengan program yang tersimpan di memorinya, dan menghasilkan output berupa informasi."

2.1.2. Pengertian Sistem

Sutabri (2012:2), "sistem adalah suatu jaringan kerja prosedur-prosedur yang saling berhubungan, berkumpul bersama-sama untuk melakukan suatu kegiatan atau menyelesaikan suatu sasaran tertentu."

Sutabri (2012:2), "sistem adalah kumpulan elemen yang berinteraksi untuk mencapai suatu tujuan tertentu."

Dari kedua pengertian tersebut, dapat disimpulkan bahwa sistem adalah suatu jaringan kerja yang saling berkerjasama untuk mencapai suatu tujuan tertentu.

2.1.3. Karakteristik Sistem

Sutabri (2012:20), menjelaskan karakteristik sistem adalah sebagai berikut :

a. Komponen Sistem *(Components)*

Suatu sistem terdiri dari sejumlah komponen yang saling berinteraksi, artinya saling berkerjasama membentuk satu kesatuan.

b. Batasan Sistem (*boundary)*

Ruang lingkup sistem merupakan daerah yang membatasi antara sistem dengan sistem yang lain atau sistem dengan lingkungan luarnya.

c. Lingkungan Luar Sistem *(Environtment)*

Bentuk apapun yang ada di luar ruang lingkup atau batasan sistem yang mempengaruhi operasi sistem disebut lingkungan luar sistem.

d. Penghubung Sistem *(interface)*

Media yang menghubungkan sistem dengan subsistem lain disebut penghubung sistem atau *interface*.

e. Masukan Sistem *(Input)*

Energi yang di masukkan ke dalam sistem disebut masukkan sistem, yang dapat berupa pemeliharaan *(maintenance input)*

f. Keluaran Sistem *(Output)*

Hasil energi yang diolah dan diklasifikasikan menjadi keluaran yang berguna. Keluaran ini merupakan masukan bagi subsistem yang lain seperti sistem informasi.

g. Pengolah Sistem *(Process)*

Suatu sistem dapat mempunyai suatu proses yang akan mengubah masukan menjadi keluaran, contohnya adalah sistem akuntasi.

h. Sasaran Sistem *(Objective)*

Suatu sistem memiliki tujuan dan sasaran yang pasti dan bersifat *deterministic*. Kalau suatu sistem tidak memilki sasaran maka operasi sistem tidak ada gunanya. Suatu sistem dikatakan berhasil bila mengenai sasaran atau tujuan yang telah direncanakan."

2.1.4. Klasifikasi Sistem

Sutabri (2012:22), menjelaskan "sistem dapat diklasifikasikan menjadi beberapa bagian yaitu:

a. Sistem abstrak dan sistem fisik

Sistem abstrak adalah sistem yang berupa pemikiran atau ide-ide yang tidak tampak secara fisik, misalnya sistem teologia, yaitu sistem yang berupa pemikiran hubungan antara manusia dengan tuhan, sedangkan sistem fisik merupakan sistem yang ada secara fisik, misalnya sistem komputer, sistem produksi, sistem penjualan, sistem administrasi personalia dan lain sebagainya.

b. Sistem alamiah dan sistem buatan manusia

Sistem alamiah adalah sistem yang terjadi melalui proses alam, tidak dibuat oleh manusia, misalnya sistem perputaran bumi, terjadinya siang malam, pergantian musim. Sedangkan sistem buatan manusia merupakan sistem yang melibatkan interaksi manusia dengan mesin yang disebut *human machine sistem*. Sistem informasi berbasis komputer merupakan contoh *human machine sistem* karena menyangkut penggunaan komputer yang berinteraksi dengan manusia.

c. Sistem determinasi dan sistem probabilistik

Sistem yang beroperasi dengan tingkah laku yang dapat diprediksi disebut sistem *deterministic*. Sistem komputer adalah contoh dari sistem yang tingkah lakunya dapat dipastikan berdasarkan program – program komputer yang dijalankan, sedangkan sistem yang bersifat probabilistik adalah sistem yang kondisi masa depannya tidak dapat diprediksi karena mengandung unsur *probabilistic.*"

2.1.5. Prinsip Pengembangan Sistem

Sutabri (2012:59), tahapan dalam pengembangan sistem adalah sebagai berikut:

1. Investigasi Sistem

   Manfaat dari fase penyelidikan ini adalah untuk menentukan masalah-masalah atau kebutuhan yang timbul. Hal ini memerlukan pengembangan sistem secara menyeluruh ataukah ada usaha lain yang dapat dilakukan untuk memecahkannya.

Sutabri (2012:64), tahapan proses yang terjadi pada tahap investigasi sistem adalah sebagai berikut:

1. Studi Awal

   Langkah pertama dalam proses penyusunan sistem, dimana gagasan untuk membangun sistem yang baru atau menyempurnakan sistem yang sedang berjalan, diterima dan dipelajari pada sistem yang berjalan paling awal. Secara umum, tujuan dari kegiatan studi awal ini adalah untuk memutuskan usaha-usaha penyusunan sistem guna menyelesaikan masalah-masalah sistem informasi.

2. Studi Kelayakan

   Selanjutnya sistem analis melakukan studi kelayakan untuk memperhitungkan apakah organisasi atau instansi dimana sistem tersebut dibuat dapat melanjutkan ke tahap berikutnya dalam proses pembangunan sistem atau tidak.

2. Desain Sistem

   Pada tahap ini sebagian besar kegiatan yang berorientasi ke komputer dilaksanakan. Rencana pembuatan program dilaksanakan dan juga dilakukan testing programnya.

3. Implementasi Sistem

   Tahap ini adalah prosedur yang dilakukan untuk menyelesaikan desain sistem yang ada dalam dokumen desain sistem yang disetujui dan menguji, menginstall dan memulai penggunaan sistem baru dan sistem yang diperbaiki.

2.1.6. Pengertian Informasi

Sutabri (2012:22), “informasi adalah data yang telah diklasifikasikan atau diolah atau diinterprestasikan untuk digunakan dalam proses pengambilan keputusan.”

Kristanto (2008:7), “informasi adalah kumpulan data yang diolah menjadi bentuk yang lebih berguna dan lebih berarti bagi yang menerima.”

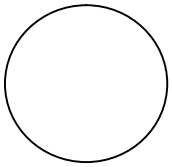
2.2. Teori Khusus

2.2.1. Pengertian DFD (*Data Flow Diagram*)

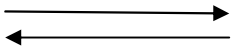
Shalahuddin et.al, (2013:70), " *Data Flow Diagram* (DFD) atau dalam bahasa indonesia menjadi Diagram Alir Data (DAD) adalah representasi grafik yang menggambarkan aliran informasi dan transformasi informasi yang diaplikasikan sebagai data yang mengalir dari masukan *(input)* dan keluaran *(output).*

Shalahuddin et.al, (2013:71), menjelaskan notasi pada DFD adalah sebagai berikut :

Tabel 2.1 Simbol-simbol *Data Flow Diagram*

No	Simbol	Keterangan
1.		Proses atau fungsi atau prosedur; pada pemodelan perangkat lunak yang akan diimplementasikan dengan pemrograman terstruktur, maka pemodelan notasi inilah yang harusnya menjadi fungsi atau prosedur di dalam kode program
2.		*File* atau basis data atau penyimpanan (*storage);* pada pemodelan perangkat lunak yang akan diimplementasikan dengan pemrograman terstruktur, maka pemodelan notasi inilah yang harusnya dibuat menjadi tabel-tabel basis data yang dibutuhkan, tabel-tabel ini juga harus sesuai dengan perancangan tabel-tabel pada basis data (*Entity Relationalship Diagram* (ERD), *Conceptual Data Model(CDM), Physical Data Model* (PDM))

Lanjutan Tabel 3.1 Simbol-simbol *Data Flow Diagram*

No.	Simbol	keterangan
3.		Entitas luar (*external entity)* atau masukan (*input)* atau keluaran (*output)* atau orang yang memakai/berinteraksi dengan perangkat lunak ynag dimodelkan atau sistem lain yang terkait denga aliran data dari sistem yang dimodelkan catatan : nama yang digunakan pada masukan (*input)* atau keluaran *(output)* berupa kata benda
4.		Aliran data; merupakan data yang dikirim antar proses, dari penyimpanan ke proses, atau dari proses ke masukan (*input)* atau keluaran *(output)* catatan : nama yang digunakan pada aliran dara biasanya berupa kata benda, dapat diawali dengan kata data misalnya “data siswa” atau tanpa kata data misalnya “siswa”

Shalahuddin et.al (2013:72) menjelaskan tahapan-tahapan perancangan dengan menggunakan DFD sebagai berikut:

1. *DFD Level* 0 atau sering disebut juga *Context Diagram*.
   DFD Level 0 menggambarkan sistem yang akan dibuat sebagai suatu entitas tunggal yang berinteraksi dengan orang maupun sistem lain. *DFD Level* 0 digunakan untuk menggambarkan interaksi antara sistem yang akan dikembangkan dengan entitas luar.

2. *DFD Level* 1

   DFD Level 1 digunakan untuk mengambarkan modul-modul yang ada dalam sistem yang akan dikembangkan. *DFD Level* 1 merupakan hasil *breakdown DFD level* 0 yang sebelumnya sudah dibuat.

3. *DFD Level* 2

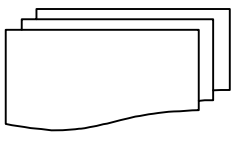
   Modul-modul pada *DFD Level* 1 dapat di-*breakdown* menjadi *DFD Level* 2. Modul mana saja yang harus di-*breakdown* lebih detail tergantung pada tingkat kedetailan modul tersebut. Apabila modul tersebut sudah cukup detail dan rinci maka modul tersebut sudah tidak perlu untuk di-*breakdown* lagi. Untuk sebuah sistem, jumlah *DFD Level* 2 sama dengan jumlah modul pada *DFD Level* di atasnya. *Breakdown* pada *level* 3,4,5, dan seterusnya aturannya sama persis dengan *DFD Level* 1 atau *Level* 2.

2.2.2. Pengertian *Blockchart*

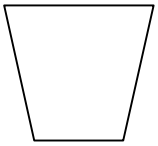
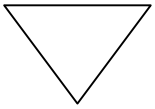
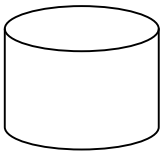
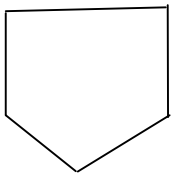
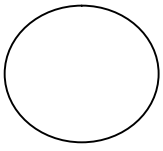
Kristanto (2008:75) menjelaskan, "*Blockchart* berfungsi untuk memodelka masukkan, keluaran, proses maupun transaksi dengan menggunakan simbol-simbol tertentu. Pembuatan *Blockchart* harus memudahkan bagi pemakai dalam memahami alur dari sestem atau transaksi."

Kristanto (2008:75) juga menjelaskan tentang simbol-simbol yang sering digunakan dalam *blockchart* dapat dilihat pada tabel berikut ini:

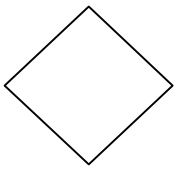
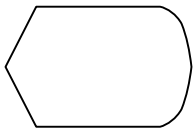
Tabel 2.2. *Simbol-simbol dalam Block Chart*

No	Simbol	Keterangan
1.		Menandakan dokumen, bisa dalam bentuk surat, formulir, buku/bendel/berkas atau cetakan
2.		Multi dokumen

Lanjutan Tabel 2.2. *Simbol-simbol dalam Block Chart*

No	Simbol	Keterangan
3.		Proses manual
4.		Proses yang dilakukan oleh komputer
5.		Menandakan dokumen yang diarsifkan (arsif manual)
6		Data penyimpanan (*data storage*)
7		Proses apa saja yang tidak terdefinisi termasuk aktifitas fisik
8		Terminasi yang mewakili simbol tertentu untuk digunakan pada aliran lain pada halaman yang lain
9		Terminasi yang mewakili simbol tertentu untuk digunakan pada aliran lain pada halaman yang sama
10		Terminasi yang menandakan awal dan akhir dari suatu aliran

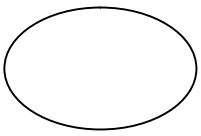
Lanjutan Tabel 2.2. *Simbol-simbol dalam Block Chart*

No	Simbol	Keterangan
11		Pengambilan keputusan *(decision)*
12		Layar peraga *(monitor)*
13		Pemasukan data secara manual

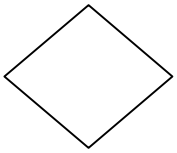
2.2.3. Pengertian *ERD* (*Entity Relational Diagram*)

Shalahuddin et.al (2013:50) menjelaskan bahwa *Entity Relationship Diagram* (*ERD*) digunakan untuk pemodelan basis data relasional. *ERD* memiliki beberapa aliran notasi seperti notasi *Chen* (dikembangkan oleh Peter Chen), *Barker* (dikembangkan oleh Richard Barker), Ian (dikembangkan oleh Palmer, Harry Ellis), notasi *Crow's Foot*, dan beberapa notasi lainnya. Namun yang banyak digunakan adalah notasi dari *Chen*. Berikut adalah simbol-simbol yang digunakan pada *ERD* dengan notasi *Chen*:

Tabel 2.3. *Simbol-simbol Entity Relationship Diagram (ERD)*

No	Simbol	Keterangan
1.		Entitas merupakan data inti yang akan disimpan
2.		Atribut atau *field* atau kolom data yang butuh disimpan dalam suatu entitas

Lanjutan Tabel 2.3. *Simbol-simbol Entity Relationship Diagram (ERD)*

No	Simbol	Keterangan
3.		Relasi, menghubungkan antara entitas dan biasanya diawali dengan kata kerja
4.		Asosiasi, penghubung antara relasi dan entitas dimana di kedua ujungnya memiliki *multiplicity* kemungkinan jumlah pemakaian. Kemungkinan jumlah maksimum keterhubungan antara entitas satu dengan entitas yang lain disebut dengan kardinalitas. Misalkan ada kardinalitas 1 ke N atau sering disebut dengan *one to many*

2.2.4. Pengertian *Flowchart*

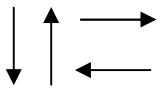
Saputra (2013:120) menjelaskan, "*Flowchart* merupakan suatu diagram yang menggambarkan alur kerja suatu sistem."

Simbol-simbol yang digunakan dapat dibagi menjadi 3 (tiga) kelompok, yakni sebagai berikut:

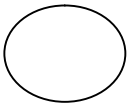
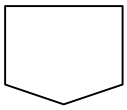
1. Flow direction symbols

Digunakan untuk menghubungkan simbol satu dengan yang lain. Berikut simbol *Flow direction symbols* seperti pada tabel dibawah ini:

Tabel 2.4. *Simbol-simbol Flow direction symbols*

No	Simbol	Nama Simbol	Keterangan
1.		**Garis Alir (*Flow Line*)**	Menyatakan jalannya arus suatu proses
2		***Communication Link***	Digunakan untuk memberikan nilai awal pada suatu variabel atau *counter*

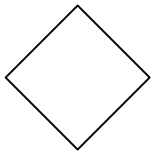
Lanjutan Tabel 2.4. *Simbol-simbol Flow direction symbols*

No	Simbol	Nama Simbol	Keterangan
3		***Connector***	Digunakan untuk menunjukkan hubungan arus proses yang terputus masih dalam halaman yang sama
4		***Offline Connector***	Digunakan untuk menunjuk-kan hubungan arus proses yang terputus masih dalam halaman yang sama

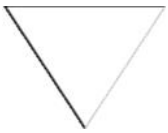
2. Processing symbols

Menunjukan jenis operasi pengolahan dalam suatu proses / prosedur. Berikut simbol *Processing symbols* seperti pada tabel dibawah ini:

Tabel 2.5. *Simbol-simbol Processing symbols*

No	Simbol	Nama Simbol	Keterangan
1.		***Process***	Menyatakan suatu tindakan (proses) yang dilakukan oleh komputer
2.		***Manual***	Menyatakan suatu tindakan (proses) yang tidak dilakukan oleh komputer
3.		***DECISION***	Menunjukkan suatu kondisi tertentu yang akan menghasilkan dua kemungkinan jawaban: ya/tidak

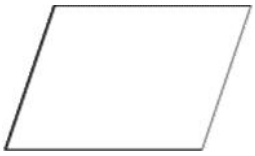
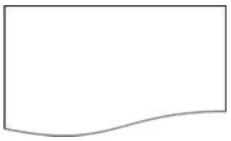
Lanjutan Tabel 2.5. *Simbol-simbol Processing symbols*

No	Simbol	Nama Simbol	Keterangan
4.		***Predefined Process***	Menyatakan penyediaan tempat penyimpanan suatu pengolahan untuk memberi harga awal
5.		***Terminal***	Menyatakan permulaan atau akhir suatu program
6.		***Keying Operation***	Menyatakan segal jenis operasi yang diproses dengan menggunakan suatu mesin yang mempunyai *keyboard*
7.		***Offline Storage***	Menunjukkan bahwa data dalam simbol ini akan disimpan ke suatu media tertentu
8.		***Manual Input***	Memasukkan data secara manual dengan menggunakan online keyboard

3. Input / Output symbols

Menunjukkan jenis peralatan yang digunakan sebagai media *input* atau *output*.

Tabel 2.6. Simbol-simbol *Input / Output symbols*

No.	Simbol	Nama Simbol	Keterangan
1.		***Input/Output***	Menyatakan proses input atau output tanpa tergantung jenis peralatannya
2.		***Punched Card***	Menyatakan *input* berasal dari kartu atau *output* ditulis ke kartu
3.		***Magnetic Tape***	Menyatakan *input* berasal dari pita magnetis atau *output* disimpan ke pita magnetis
4.		***Disk Storage***	Menyatakan *input* berasal dari dari *disk* atau *output* disimpan ke *disk*
5.		***Document***	Digunakan untuk mencetak keluaran dalam bentuk dokumen (melalui *printer*)
6.		***Display***	Digunakan untuk mencetak keluaran dalam layar *monitor*

2.2.5. Pengertian Kamus Data

Shalahuddin et.al (2013:73), ”Kamus data adalah kumpulan dafar elemen data yang mengalir pada sistem perangkat lunak sehingga masukan (*input*) dan keluaran (*output*) dapat dipahami secara umum (memiliki standar cara

penulisan)."

Shalahuddin et.al (2013:74) menjelaskan bahwa kamus data memiliki beberapa simbol untuk menjelaskan informasi tambahan, yaitu sebagai berikut:

Table 2.7. *Simbol-simbol dalam Kamus Data*

No	Simbol	Arti
1	=	disusun atau terdiri atas
2	+	*AND* atau dan
3	( )	data opsional
4	[|]	baik...atau...
5	*...*	batas komentar
6	{}n	n kali diulang / bernilai banyak

2.3. Teori Judul

2.3.1. Pengertian Website

Asropudin (2013:109), "*web* adalah sebuah kumpulan halaman (*webpage*) yang diawali dengan halaman muka (*homepage*) yang berisikan informasi, iklan, serta program interaksi."

Salim (2013:513), "*web* adalah sistem untuk mengakses, memanipulasi, dan mengunduh dokumen hepertaut yang terdapat dalam komputer yang dihubungkan melalui *internet*."

2.3.2. Pengertian *e-news*

E-news berarti berita atau informasi yang dikemas secara menarik dan disampaikan melalui media ekeltronik secara online, yang biasanya ditampilkan oleh komputer yang terhubung dengan internet / ditampilkan oleh browser yang terhubung dengan sebuah server. (http://dickysyuhada.blogspot.com/2012/11/apa-itu-e-newse-governmente-learninggps.html)

E- News adalah berita yang menggunakan perangkat komputer yang terhubung dengan bejaringan komputer global. terdapat juga beberapa istilah

untuk berita elektronik atau *electronic news* diantaranya *online news* atau *digital news*.(http://alistadiponegoro.blogspot.com/2009/07/e-news-definisi-e-news-e-news-adalah.html)

2.3.3. Pengertian Website Informasi *e-news* pada Harian Umum Prabumulih Pos

Merupakan sebuah web yang berisi kumpulan berita yang disampaikan melalui media elektronik secara online yang dapat memudahkan pelanggan Harian Umum Prabumulih Pos untuk mendapatkan berita ter-*update*.

2.4. Teori Program

2.4.1. Basis Data (*Database*)

Priyadi (2014:2), "Secara umum, pengertian basis data adalah sekumpulan fakta berupa representasi tabel yang saling berhubungan dan disimpan dalam media penyimpanan secara digital."

Priyadi (2014:3) juga menjelaskan beberapa elemen basis data, di antaranya adalah sebagai berikut:

1. Tabel

   Pada suatu basis data, tabel di representasikan menjadi suatu bentuk segiempat berupa matriks, yang terdiri dari kolom dan baris.

2. *Field*

   Pada tabel basis data, kolom merupakan representasi untuk sebuah nama *field* yang pembacaan datanya dilakukan secara vertikal.

3. *Record*

   Pada tabel basis data, baris merupakan suatu representasi untuk sebuah *record* yang pembacaan datanya dilakukan secara horizontal. Satu baris pada sebuah tabel merupakan data yang dimiliki oleh satu *record*. Nilai-nilai yang dimiliki oleh sebuah *record* merupakan gabungan dari semua *field* yang terdapat dalam tabel tersebut.

4. Kardinalitas

   Kardinalitas merupakan batasan dari banyaknya hubungan, yang dapat dilakukan oleh himpunan entitas dalam melakukan relasi dengan himpunan entitas lainnya. Variasi kemungkinan untuk melakukan relasi yang dimiliki oleh kardinalitas terdiri dari empat macam, yaitu:

   a. Satu ke satu ⟶ (1:1).
   b. Satu ke banyak ⟶ (1:N).
   c. Banyak ke satu ⟶ (N:1).
   d. Banyak ke banyak ⟶ (N:N).

2.4.2. Sekilas Tentang *HTML*

2.4.2.1. Pengertian *HTML*

Badiyanto (2013:23), "*HTML (Hyper Text Markup Language)* adalah Bahasa standar pemrograman untuk membuat halaman *web* yang terdiri dari kode-kode *tag* tertentu, kemudian kode-kode tersebut diterjemahkan oleh *web browser* untuk menampilkan halaman *web* yang terdiri dari berbagai macam *format* tampilan seperti teks, grafik, animasi *link*, maupun *audio-video*. Berikut contoh skrip yang ada pada *HTML* :

Table 2.8. *Tag HTML*

No	*Tag HTML*	Keterangan
1	<*html*>	*Tag* Skrip *HTML*
2	<*head*>	*Tag* bagian kepala *HTML*
3	<*title*>	*Tag* untuk menempatkan judul
4	<*body*>	Skrip untuk isi *HTML*
5	<h1> sampai <h6>	*Tag* untuk menentukan ukuran *font* pada judul atau subjudul pada halaman *web*
6	<p>	*Tag* untuk pemberian paragraph
7	 	*Tag* untuk pemberian pindah baris, seperti *enter*

Lanjutan Table 2.8. *Tag HTML*

No	*Tag HTML*	Keterangan
8	<b>	*Tag* menampilkan teks dengan *font* mencetak *font* tebal (*bold*)
9	<i>	*Tag* menampilkan teks dengan *font* cetak miring (*italic*)
10	<u>	*Tag* menampilkan teks dengan *font* garis bawah (*underline*)
11	**	*Tag* menampilkan teks dengan *font* cetak menjadi tebal (mirip *bold*)
12	*<code>*	*Tag* menampilkan teks dengan *font* cetak huruf menjadi *font courier*
13	<tt>	*Tag* menampilkan teks dengan *font* cetak huruf menjadi *font teletype writer*
14	*<sampp>*	*Tag* menampilkan teks dengan *font* cetak huruf menjadi *font code*
15	*<pre>*	Memformat tampilan teks semua spasi yang bukan *script* akan sama letaknya dengan yang ditulis
16	* *	Membuat nomor *bullet*, pada baris berikutnya harus ditulis * * sebelum menulis teks
17	* *	Membuat nomor urut angka
18	*<table border>*	Gunanya untuk membuat *table*
19	*<tr>*	*Tag* membuat baris dalam *table*
20	*<td>*	*Tag* membuat kolom dalam *table*
21	*<th>*	*Tag table header*, membuat kolom judul

2.4.3. Sekilas Tentang *PHP*

2.4.3.1. Pengertian *PHP* (*Hypertext Preprocessor*)

Saputra (2013:1), "*PHP* adalah singkatan dari (*Hypertext Prepocessor*) yang merupakan suatu bahasa pemograman yang hanya dapat berjalan pada sisi *server* (*Server Side Scripting*)."

Sedangkan menurut Badiyanto (2013:32), "*PHP: Hypertext Prepocessor* adalah bahasa skrip yang dapat ditanamkan atau disisipkan ke dalam *HTML*/*PHP* banyak dipakai untuk membuat situs *web* dinamis.

Dari kedua pengertian di atas, dapat disimpulkan bahwa *PHP* adalah salah satu bahasa pemrograman dirancang untuk bekerja atau membangun sebuah aplikasi berbasis *web.*

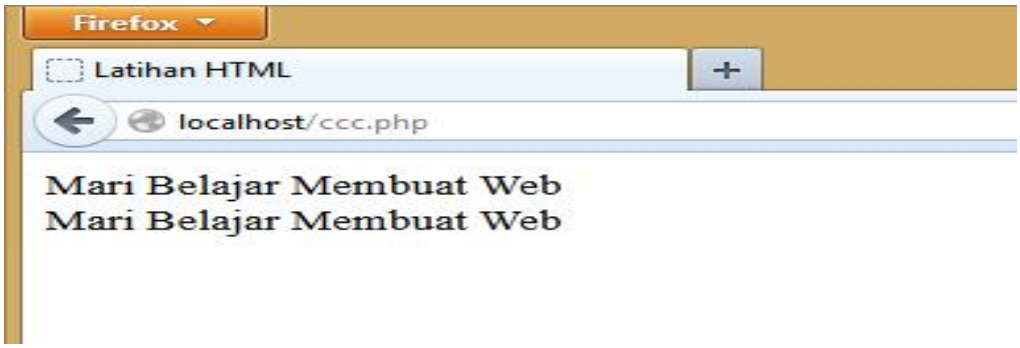
2.4.3.2. Skrip *PHP*

Badiyanto (2013:32), "menjelaskan tentang skrip *PHP*, *PHP* yang merupakan sebuah Bahasa *scripting* yang terpasang pada *HTML* dan skrip *HTML* adalah bahasa standar untuk membuat halaman *web*." Sebagai contoh, berikut adalah contoh kode *PHP* yang berada di dalam kode *HTML*:

```
<HTML>
<HEAD>
<TITLE> Latihan HTML</TITLE>
</HEAD>
<BODY>
<?php
        Printf("Mari Belajar Membuat Web");
// atau
Echo "<br>";
Echo "Mari Belajar Membuat Web";
?>
</BODY>
```

</HTML>

Bila dijalankan melalui *browser*, kode tersebut membentuk tampilan seperti berikut:



Gambar 2.1. *Tampilan kode PHP pada browser*

2.4.3.3. Tipe Data *PHP*

MADCOMS (2011:221) menjelaskan tentang tipe data *PHP*, yaitu:

1. Tipe Data *Integer*

   Tipe data ini berisikan data semua bilangan bulat yang besarnya range sama dengan data pada bahasa C, yaitu antara **-2, 147, 483, 648** sampa **+2, 147, 483, 647** pada *platform* **32bit**. Apabila data di luar kisaran tersebut maka *PHP* secara otomatis mengkonversikan data tersebut dari tipe ***Integer*** menjadi tipe ***Floating Point***.

2. Tipe Data *Floating Point*

   Tipe data ini berisikan bilangan pecahan atau bilangan desimal. Kisaran data *floating* adalah antara **1.7E-308** sampai **1.7E+308**. Bentuk dari data ini adalah desimal ataupun dalam bentuk pangkat.

3. Variabel

   Dalam program *PHP*, variabel digunakan untuk menyimpan data sementara baik jenis *string*, *integer*, maupun *array*. Variabel dinyatakan dengan tanda $ di depan nama variabel. Nama variabel dapat berupa huruf, angka maupun garis bawah. Namun dalam penulisannya, variabel harus

diawali dengan huruf atau garis bawah (_), kemudian diikuti huruf atau angka. Penulisan variabel yang diawali angka tidak dibenarkan.

2.4.4. Sekilas Tentang *MySql*

2.4.4.1. Pengertian *MySql*

Saputra (2013:14), "*MySQL* merupakan *database storage engine* yang paling banyak digunakan oleh *web developer* karena sifatnya *free*, alias gratis."

Sedangkan menurut Badiyanto (2013:57), "*MySQL* merupakan sebuah *database* server *SQL multiuser* dan *multithreaded."*

Sedangkan menurut Kadir (2008:342) menjelaskan bahwa *MySQL* adalah salah satu jenis *database server* yang sangat populer yang bersifat *open source* (Anda tidak perlu membayar untuk menggunakannya) pada bagian *platform* (kecuali untuk jenis *Enterprise*, yang bersifat komersial). *MySQL* termasuk jenis *RDBMS* (*Relational Database Management System*). Itulah sebabnya, istilah seperti tabel, baris dan kolom digunakan pada *MySQL*. Pada *MySQL*, sebuah *database* mengandung satu atau sejumlah tabel.

Dari ketiga pengertian di atas, dapat disimpulkan bahwa *MySQL* adalah jenis *software* pada *Relational Database Management System* (*RDBMS*) pada sebuah manajemen *database* sebagai basis data *server SQL*.

2.4.4.2. Fungsi-Fungsi *MySql*

Kadir (2008:360), menjelaskan tentang sejumlah fungsi yang berawalan *mysql_* yang digunakan untuk mengakses *database server MySQL* sebagai berikut:

Tabel 2.9. *Fungsi-Fungsi MySQL*

No	Nama Fungsi	Kegunaan
1	*MySQL_connect*()	Membuka koneksi ke *server MySQL*
2	*MySQL_close*()	Menutup koneksi *MySQL*
3	*MySQL_select_db*()	Membuka/mengaktifkan suatu basis data
4	*MySQL_query*()	Mengeksekusi *string query*

Lanjutan Tabel 2.9. *Fungsi-Fungsi MySQL*

No	Nama Fungsi	Kegunaan
5	*MySQL_db_query*()	Melakukan *query*
6	*MySQL_num_fields*()	Menghitung jumlah *field* dalam tabel tertentu
7	*MySQL_fecth_assoc*()	Mengambil hasil *query* sebelumnya
8	*MySQL_fecth_result*()	Membebaskan *resource* dari hasil *query*
9	*MySQL_fecth_row*()	Menampilkan hasil *query* ke dalam *array*
10	*MySQL_fecth_array*()	Menampilakan hasil *query* sebagai *array*
11	*MySQL_fecth_field*()	Menampilkan informasi *field* termasuk tipe datanya
12	*MySQL_affected_rows*()	Menampilkan jumlah baris yang dikenai operasi *SQL*
13	*MySQL_errno*	Menampilkan kode kesalahan (*error number*) dari perintah *SQL*
14	*MySQL_create_db*()	Membuat *database MySQL*
15	*MySQL_drop_db*()	Menghapus *database MySQL*
16	*MySQL_list_dbs*()	Menghasilkan seluruh daftar *database MySQL*
17	*MySQL_list_tables*()	Menampilkan seluruh tabel dalam *database* tertentu
18	*MySQL_list_fields*()	Menampilkan seluruh *field* dalam tabel tertentu

(Sumber : Kadir, 2008:360)

2.4.5. Pengertian *Sublime Text 2*

Sibarani (2013: 24), "*Sublime Text* adalah editor teks yang digunakan untuk menerjemahkan hasil analisa dan rancangan kedalam bentuk *code*."

Berikut beberapa fitur yang diunggulkan dari aplikasi *Sublime Text*:

a. *Goto Anything*

Fitur yang sangat membantu dalam membuka *file* ataupun menjelajahi isi dari *file* hanya dengan beberapa *keystrokes*.

b. *Multiple Selections*

Fitur ini memungkinkan *user* untuk mengubah secara interaktif banyak baris sekaligus, mengubah nama variabel dengan mudah, dan memanipulasi *file* lebih cepat dari sebelumnya.

c. *Command Pallete*

Dengan hanya beberapa *keystorkes*, *user* dapat dengan cepat mencari fungsi yang diinginkan, tanpa harus menavigasi melalu menu.

d. *Distraction Free Mode*

Bila *user* memerlukan fokus penuh pada aplikasi ini, fitur ini dapat membantu *user* dengan memberikan tampilan layar penuh. *Split Editing* dapatkan hasil yang maksimal dari *monitor* layar lebar dengan dukungan *editing* perpecahan. Mengedit sisi *file* dengan sisi, atau mengedit dua lokasi di satu *file*. Anda dapat mengedit dengan banyak baris dan kolom yang *user* inginkan.

e. *Instant Project Switch*

Menangkap semua *file* yang dimasukkan kedalam *project* pada aplikasi ini. Terintegrasi dengan fitur *Goto Anything* untuk menjelajahi semua *file* yang ada ataupun untuk beralih ke *file* dalam *project* lainnya dengan cepat.

f. *Plugin API*

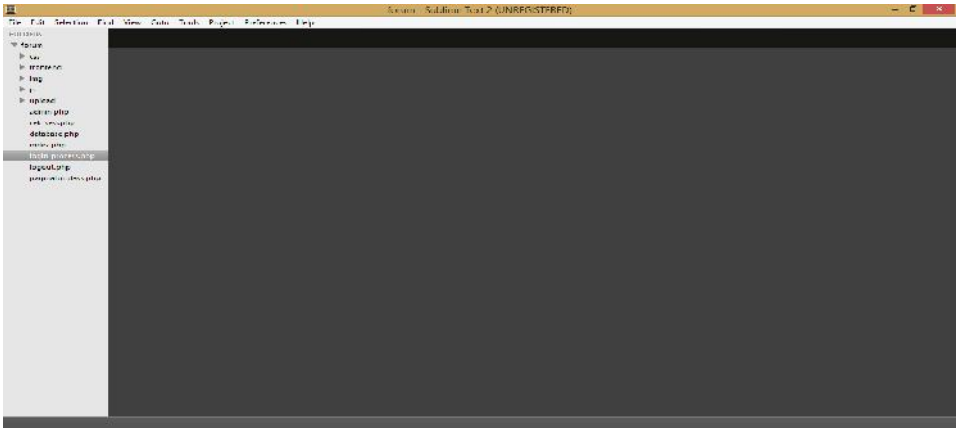
Dilengkapi dengan *plugin* API berbasis *Phyton* sehingga membuat aplikasi ini sangat tangguh.

g. *Customize Anything*

Aplikasi ini memberikan *user* fleksibilitas dalam hal pengaturan fungsional dalam aplikasi ini.

h. *Cross Platform*

Aplikasi ini dapat berjalan hampir disemua *operating system modern* seperti *Windows*, *OS X*, dan *Linux based operating system*.



Gambar 2.2. *Ruang kerja dari Sublime Text 2*

2.4.6. *JavaScript*

Badiyanto (2013:27), "*JavaScript* adalah bahasa pemrograman java, perintah-perintahnya ditulis dengan kode yang disebut skrip. *Java* adalah Bahasa pemrograman berorientasi objek, sedangkan *script* adalah serangkaian intruksi program."

Saputra (2012:3). "*JavaScript* adalah Bahasa *scripting* yang handal yang berjalan pada sisi *client. JavaScript* merupakan sebuah Bahasa *scripting* yang dikembangkan oleh *Netscape*.

Dari istilah di atas, dapat disimpulkan bahwa *JavaScript* adalah Bahasa pemrograman yang memiliki sekumpulan perintah khusus yang digunakan untuk membuat sebuah halaman web yang lebih responsif dan interaktif.

2.4.7. *CSS (Cascading Style Sheets)*

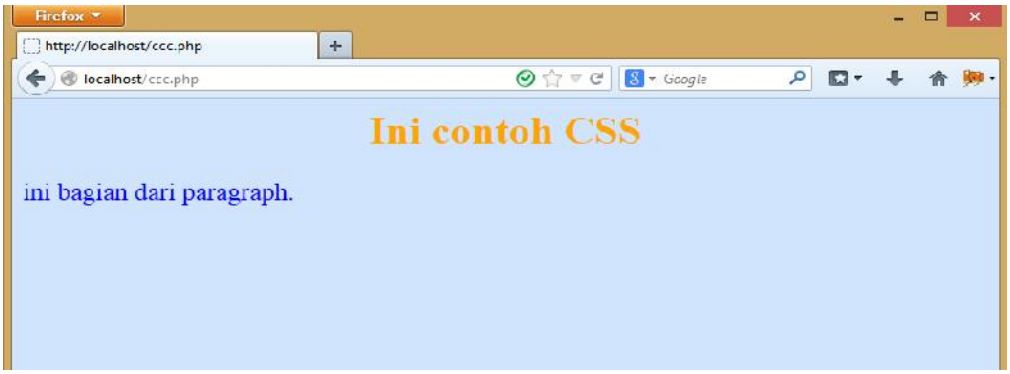
Badiyanto (2013:25), "*CSS* adalah kepanjangan dari *Cascading Style Sheets*, *CSS* skrip yang berisi rangkaian intruksi yang memantau suatu teks akan tertampil dihalaman *web browser*." Contoh cara penggunaan *CSS* sebagai berikut :

<html>

<head>

```
<style>
body {
background-color: #d0e4fe;
}
h1{
color: orange;
text-align: center;
}
p{
font-family: "Times New Roman";
font-size: 20px;
color: blue;
}
</style>
</head>
<body>
<h1> Ini contoh CSS </h1>
<p>ini bagian dari paragraph.</p>
</body>
</html>
```

Bila dijalankan melalui *browser*, kode tersebut membentuk tampilan seperti berikut:



Gambar 2.3. *Tampilan kode CSS*